Computers and Mathematics with Applications 61 (2011) 1129–1144

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

An improved FCMBP fuzzy clustering method based on evolutionary programming☆

Qing Tan [a,b], Qing He [a,*], Weizhong Zhao [a,b], Zhongzhi Shi [a], E.S. Lee [c]

[a] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[b] Graduate School, Chinese Academy of Sciences, Beijing 100049, China
[c] Department of Industrial Engineering, Kansas State University, Manhattan, KS 66506, USA

ARTICLE INFO

Article history:
Received 27 September 2010
Received in revised form 24 December 2010
Accepted 27 December 2010

Keywords:
Fuzzy clustering
FCMBP fuzzy clustering
Optimal fuzzy equivalent matrix
Evolutionary programming

ABSTRACT

In current PC computing environment, the fuzzy clustering method based on perturbation (FCMBP) is failed when dealing with similar matrices whose orders are higher than tens. The reason is that the traversal process adopted in FCMBP is exponential complexity. This paper treated the process of finding fuzzy equivalent matrices with smallest error from an optimization point of view and proposed an improved FCMBP fuzzy clustering method based on evolutionary programming. The method seeks the optimal fuzzy equivalent matrix which is nearest to the given fuzzy similar matrix by evolving a population of candidate solutions over a number of generations. A new population is formed from an existing population through the use of a mutation operator. Better solutions survive into next generation and finally the globally optimal fuzzy equivalent matrix could be obtained or approximately obtained. Compared with FCMBP, the improved method has the following advantages: (1) Traversal searching is avoided by introducing an evolutionary programming based optimization technique. (2) For low-order matrices, the method has much better efficiency in finding the globally optimal fuzzy equivalent matrix. (3) Matrices with hundreds of orders could be managed. The method could quickly get a more accurate solution than that obtained by the transitive closure method and higher precision requirement could be achieved by further iterations. And the method is adaptable for matrices of higher order. (4) The method is robust and not sensitive to parameters.

1. Introduction

The concept of fuzzy clustering analysis was firstly carried out by Ruspini in 1969 [1]. One of the fuzzy clustering methods is fuzzy k-means clustering algorithm [2]. Another method in common use is transforming a fuzzy similar matrix R into a fuzzy equivalent $R^* = t(R)$ by finding the transitive closure of R; the final clustering is then made by R^*. However, since the process of finding the transitive closure of R consists of making a series of transformations, it is not theoretically assured whether the clustering result obtained by R^* really reflects the original clustering problem [3]. In order to deal with this lack of fidelity problem, Li et al. [4] proposed the basic theory of resolution and parameter system then put forward a fuzzy clustering method based on perturbation (FCMBP), whose main idea is to try to find a fuzzy equivalent $R^{\#}$, which is closest to R by a certain distance. Thereafter, FCMBP is further studied and several important results are obtained in paper [5–8]. [5] proposed the concept of fuzzy equivalent normal form in order to express the solutions. Also the uniqueness of the

☆ This work is supported by the National Natural Science Foundation of China (No. 60933004, 60975039, 61035003, 60903141, 61072085), National Basic Research Priorities Programme (No. 2007CB311004) and National Science and Technology Support Plan (No. 2006BAC08B06).
* Corresponding author. Tel.: +86 010 62600542.
E-mail address: heq@ics.ict.ac.cn (Q. He).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.12.063

parameter systems corresponding to standard resolution process was pointed out and the existence of optimal fuzzy equivalence matrix was proved. In paper [6], based on the theory of distribution structure of fuzzy similar matrix equation $X^2 = X$, the existence theorem and local uniqueness theorem of the optimal fuzzy equivalent matrix have been proved by using the method of abstract analysis. And also the relation between local optimal fuzzy equivalent matrix and the given fuzzy similar matrix had been pointed out. [7] proved that the FCMBP fuzzy clustering methods have smaller error than transitive closure methods and gave examples to show their clustering results are not always the same. [8] provided a systemic and detailed discussion about the theory of FCMBP fuzzy clustering method.

The FCMBP fuzzy clustering method in paper [8] provides a way to obtain the optimal fuzzy equivalent matrix. However, it needs to run over all the combination possibilities of all classes of similar equivalent standard forms and all elements in a symmetric group S_n. The number of the equivalent matrices to be calculated grows exponentially as the order of the raw similar matrix rises. It is computationally intractable when n, the order of the matrix, is high. For example, the method needs to run over 2.3×10^{11} possibilities when $n = 12$, and this number grows to 6.7×10^{15} and 7.8×10^{23} when $n = 15$ and $n = 20$ correspondingly. So the method is out of work in current PC computing environment when dealing with high-order similar matrices.

In order to make this exponential complexity algorithm still work on high-order situations, the traversal process adopted in FCMBP must be break up. In this paper, we consider the problem of finding the optimal fuzzy equivalent matrix from an optimization point of view. The optimization objective is to make the distance between the obtained fuzzy equivalent matrix $\overline{R}$ and the raw fuzzy similar matrix R as small as possible, viz. make the lack of fidelity brought about by clustering according to $\overline{R}$ as less as possible.

Traditional optimization techniques like gradient descent algorithm or direct, analytical methods have the advantages of high computation efficiency and strong reliability. However, these methods always have strict constraint requirements on the objective function such as single peak demands and continuously differentiable requirements. For the problem of finding optimal fuzzy equivalent matrix, its objective function does not satisfy the single peak demands and even could not be expressed as a corresponding expression. Therefore these traditional optimization methods cannot be applied to the problem.

Evolutionary computation is a kind of useful method of optimization when other optimization methods fail in finding the optimal solution [9]. It is suitable for difficult combinatorial and real-valued function optimization problems in which the fitness landscapes are rugged and have many locally optimal solutions. These methods do not depend on the first- and second-differentials of the objective function of the problem to be optimized and even have no demands on whether the objective function has an explicit expression. All these features make it very suitable for solving the proposed optimization problem. As a typical evolutionary algorithm, evolutionary programming (EP) [10] is a heuristic method based on simulating the mechanics of natural selections. Compared with genetic algorithm (GA) [11], EP does not involve encoding the problem solutions as fixed-length binary strings but directly represents them according to the problem to be optimized. Specific to the problem of finding optimal fuzzy equivalent matrix, feasible solutions are represented by two factors: $\sigma \in S_n$ where S_n is a symmetric group and $\tilde{\tilde{X}} \in \tilde{X}_n/\approx$ where $\tilde{X}_n/\approx$ stores all classes of similar equivalent standard forms. Moreover, EP applies mutation operators only while the classical GA uses crossover, mutation and other genetic operators. This mechanism avoids the difficulty of defining a reasonable crossover operator in the proposed optimization problem. For the above reasons, we introduce evolutionary programming based optimization technique to find the optimal fuzzy equivalent matrix and thus propose an EP based FCMBP (EP-FCMBP) fuzzy clustering method. The method seeks the optimal solution by evolving a population of candidate fuzzy equivalent matrices over a number of generations. A new population is formed from an existing population through the use of a mutation operator. Through the use of a competition scheme, matrices with a relatively less lack of fidelity have a greater chance of survival than the poorer solutions, which guarantees the population evolves towards the global optimal point.

The rest of this paper is organized as follows: Section 2 introduces the basic idea of FCMBP and gives analysis of its computational complexity. Then in Section 3, we present our EP-FCMBP fuzzy clustering method in detail. Experimental results and analysis are given in Section 4, followed by our conclusions in Section 5.

2. Summary of FCMBP fuzzy clustering method

We shall follow the notations used in the earlier paper [8]. The main results in [8] are briefly summarized in the following.

Let Y_n be the set of n-order fuzzy similar matrices, X_n be the set of n-order fuzzy equivalent matrices, and S_n be a symmetric group. Then we have the following propositions.

Proposition 1 ([4]). *For any $X \in Y_n$, we have*

$$X \in X_n \Leftrightarrow x_{ik} \wedge x_{kj} \le x_{ij}, \quad \forall 1 \le i \ne j \ne k \le n. \tag{1}$$

Proposition 2 ([4]). *For any $\sigma \in S_n$, $X = (x_{ij})_{n\times n}$, we have the following.*

(1) *If $X \in Y_n$, then $X_\sigma = (x_{\sigma(i)\sigma(j)})_{n\times n} \in Y_n$.*
(2) *If $X \in X_n$, then $X_\sigma = (x_{\sigma(i)\sigma(j)})_{n\times n} \in X_n$.*
(3) *If $X \in X_n$, then $X(I_m) \in X_m$ and $X(I_m^c) \in X_{n-m}$.*

(4) *Let $X \in Y_n$. If there exist a $t \in [0, 1]$ and X has a resolution satisfying the following conditions:*
   (a) $X(I_m) \in X_m$ *and* $X(I_m^c) \in X_{n-m}$*;*
   (b) $X(I_m) \geq t$ *and* $X(I_m^c) \geq t$*;*
   (c) $X(I_m, I_m^c) = (t)_{m\times(n-m)}$ *and* $X(I_m^c, I_m) = (t)_{(n-m)\times m}$*;*
   then $X \in X_n$ and we say that X has a resolution structure, where $X(I_m, I_m^c) = (x_{ij})_{m\times(n-m)}$ denotes the matrix consisted of the elements of x_{ij}, $i \in I_m$, $j \in I_m^c$, and others are similarly defined.
(5) *If $X \in Y_n$, then $X \in X_n \Leftrightarrow X$ has a resolution structure.*
(6) *The solution of an n-order matrix equation $X^2 = X$ can be represented by $n - 1$ parameters.*

For any $X \in X_n$, which has obtained the resolution structure for X, we have $X(I_m) \in X_m$ and $X(I_m^c) \in X_{n-m}$ and call these the first resolution structure. Then, for $X(I_m)$ and $X(I_m^c)$, we can similarly obtain the resolution structure of $X(I_m)$ and $X(I_m^c)$. This process is continued until the submatrices become one-order submatrices. Since every resolving process is carried out according to some parameter t, we can illustrate the resolving process by a diagram as follows:

$$\begin{array}{lcl} X(I_m) : t_2 & & t_2 \\ \uparrow & \text{simplified as:} & \uparrow \\ X : t_1 \to X(I_m^c) : t_3, & & t_1 \to t_3. \end{array}$$

After the matrix has been completely resolved in a step-by-step fashion, we obtain a completed parameter system of X. The resolution structure of X obtained by the following process is called a standard resolution of X, and the process is called a standard resolution process [4].

(1) Let $t = \wedge\{x_{ij} : 1 \leq i \neq j \leq n\}$.
(2) If $t = 1$, then $X = (1)_{n\times n}$ and let $I_m = \{1\}$ and $I_m^c = \{2, 3, \dots, n\}$; if $t < 1$, then find the first column (or row) which contains the most t. Suppose that the column is the j_0th column. Let $x_{i_1j_0} = x_{i_2j_0} = \cdots = x_{i_{n-m}j_0} = t$ $(i_1 < i_2 < \cdots < i_{n-m})$, and let $I_m^c = \{i_1, i_2, \dots, i_{n-m}\}$ and $I_m = \{1, 2, 3, \dots, n\} \setminus I_m^c$.
(3) Carry out the resolution structure by referring to I_m and I_m^c defined in above step. Thus, the first standard resolution structure $(X(I_m), X(I_m^c), X(I_m, I_m^c), X(I_m^c, I_m))$ was obtained.

In X_n, we define an equivalent relation "$\sim$" as follows: $\forall X, Y \in X_n, X \sim Y \Leftrightarrow \exists\sigma \in S_n$, s.t. $Y = X_\sigma$.

Definition 1 (*[8]*)**.** Let $X \in X_n$, X is called a fuzzy equivalent standard form if its lower triangular matrix has the following form of blocks.

(1) For each block, all the elements of the block are equal and the up-right elements are closest to the diagonal elements 1.
(2) The elements in the upper blocks are bigger than the ones in the lower blocks.
(3) The elements in the right blocks are bigger than or equal to the ones in the left blocks.
(4) Each block has more rows than columns or the rows and columns are equal.
(5) The lower triangular is divided into $n - 1$ blocks.

Proposition 3 (*[8]*)**.** *Let $X \in X_n$. If X has a standard resolution structure $(X(I_m), X(I_m^c), X(I_m, I_m^c), X(I_m^c, I_m))$, then the following block statements are true.*

(1) *$\exists\sigma \in S_n$, s.t. X_σ has the following block matrix form:*

$$X_\sigma = \begin{bmatrix} X(I_m) & X(I_m, I_m^c) \\ X(I_m^c, I_m) & X(I_m^c) \end{bmatrix}. \tag{2}$$

(2) $X(I_m, I_m^c) = (t)_{m\times(n-m)}$, $X(I_m^c, I_m) = (t)_{(n-m)\times m}$.
(3) *If $t = 1$, then $X = (1)_{n\times n}$. If $t < 1$, then*
   (a) $m \leq n - m$,
   (b) $X(I_m) > (t)_{m\times m}$,
   (c) $X(I_m^c) \geq (t)_{(n-m)\times(n-m)}$, *or there exist t in $X(I_m^c)$ and $s \leq n - 2m$, where s is the number of t in the column of $X(I_m^c)$ which has the most t.*
   (d) *The up-right element of $X(I_m^c, I_m)$ is nearest to the diagonal element 1.*

Theorem 1 (*[8]*)**.** *For any $X \in X_n$, $\exists\sigma \in S_n$ s.t. X_σ is a fuzzy equivalent standard form.*

Theorem 2 (*[8]*)**.** *For any $X, Y \in X_n$, $X \sim Y \Leftrightarrow X$ and Y have the same equivalent standard form.*

Let $X \in X_n$, and $[X]$ is the equivalent class of X. Then $[X]$ can be represented by the equivalent standard form.

Theorem 3 (*[8]*)**.** *If $X \in X_n$, then $X \sim Y \Leftrightarrow X$ and Y have the same standard parameter system.*

Two standard parameter systems are called similar if they have the same diagram, but their parameters may not be equal. We introduce another relation "$\approx$" in X_n as: $X \approx Y \Leftrightarrow X$ and Y have similar standard parameter systems, whose relation is an equivalent relation in X_n and will be called a translation equivalent relation. All of the equivalent standard forms together form a set denoted by $\tilde{X}_n$. It is obvious that $|\tilde{X}_n| = |X_n/\sim|$. Obviously, "$\approx$" is also an equivalent relation in $\tilde{X}_n$.

Let J be the set of all the standard parameter systems. It is obvious that there exists a one-to-one correspondence between J and $\tilde{X}_n$. So the equivalent relation "$\approx$" can be transplanted into J. Let $X(T)$ denote the equivalent standard form which has the standard parameter system T.

Theorem 4 ([8]). $X_n = \bigcup_{\sigma\in S_n} \bigcup_{\tilde{\tilde{T}}\in J/\approx} \bigcup_{S\in\tilde{\tilde{T}}} \{X(S)_\sigma\}$.

Denote $C(\tilde{\tilde{T}}) = \{S$: *S is a parameter system that has the same diagram as T, but its inequalities may not be necessarily strictly* $\}$. $X = X(T)$, $C(\tilde{\tilde{X}}) = \{X(S) : S \in C(\tilde{\tilde{T}})\}$ *is the set of fuzzy equivalent matrices corresponding to* $C(\tilde{\tilde{T}})$.

Theorem 5 ([8]). $X_n = \bigcup_{\sigma\in S_n} \bigcup_{\tilde{\tilde{T}}\in J/\approx} \bigcup_{S\in C(\tilde{\tilde{T}})} \{X(S)_\sigma\}$.

Definition 2 ([8]). The distance between fuzzy similar matrix A and B is defined as:

$$d(A, B) = \sqrt{2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(a_{ij} - b_{ij})^2} \quad A, B \in Y_n. \tag{3}$$

Theorem 6 ([8]). *For any $R \in Y_n$, there exists $R^\# \in X_n$ s.t.*

$$d(R^\#, R) = \inf_{X\in X_n} d(X, R). \tag{4}$$

The fuzzy equivalent matrix that is closest to the given similar matrix is called a globally optimal fuzzy equivalent matrix. Obviously, it has the least lack of fidelity brought about by clustering according to the globally optimal fuzzy equivalent matrix.

Corollary 1 ([8]). *Let $R \in Y_n$ and $R^\#$ be the optimal equivalent matrix of R. Then* $R \in X_n \Leftrightarrow R^\# = R \Leftrightarrow d(R, R^\#) = 0$.

Corollary 2 ([8]). *$\forall R \in Y_n$, let $R^\#$ be the optimal fuzzy equivalent matrix of R and R^* be the transitive closure of R. Then* $d(R, R^\#) \le d(R, R^*)$.

Corollary 3 ([8]). *$\forall R \in Y_n$, let $R^\#$ be the optimal fuzzy equivalent matrix of R and R^* be the transitive closure of R. Then* $R \in X_n \Leftrightarrow R^* = R^\# = R$.

Theorem 7 ([8]). *Let $R \in Y_n$. If there exists $R^\# \in C(\tilde{\tilde{X}}) \subseteq X_n$ s.t. $d(R, R^\#) = \inf_{X\in C(\tilde{\tilde{X}})} d(X, R)$, then $t_i = \frac{b_{i1}+b_{i2}+\cdots+b_{im_i}}{m_i}$ $(i = 1, 2, \ldots, n-1)$, where $t_1, t_2, \ldots, t_{n-1}$ are parameters in the parameter system of $R^\#$ and $b_{i1}, b_{i2}, \ldots, b_{im_i}$ are elements in R corresponding to t_i $(i = 1, 2, \ldots, n-1)$ in $R^\#$.*

Based on the above result, a FCMBP method for calculating the globally optimal fuzzy equivalent matrix could be obtained as follows [8].

Step 1. Construct a storage of all classes of similar equivalent standard forms, $\tilde{X}_n/\approx$ and their parameter systems, $J/\approx$.
Step 2. Let $R \in Y_n$.
Step 3. Let $\sigma \in S_n$.
Step 4. Let $\tilde{\tilde{X}} \in \tilde{X}_n/\approx$.
Step 5. Calculate R_σ.
Step 6. Find $b_{i1}, b_{i2}, \ldots, b_{im_i}$ $(i = 1, 2, \ldots, n-1)$ in R_σ corresponding to t_i in $\tilde{\tilde{X}}$.
Step 7. Calculate $t_i' = \frac{b_{i1}+b_{i2}+\cdots+b_{im_i}}{m_i}$, $i = 1, 2, \ldots, n-1$.
Step 8. Check whether t_i' $(i = 1, 2, \ldots, n-1)$ satisfies the inequalities given by $\tilde{\tilde{X}}$. If not, then go to Step 4. If they are, then go to the next step.
Step 9. Construct a matrix X' by using t_i' $(i = 1, 2, \ldots, n-1)$ such that $X' \in \tilde{\tilde{X}}$ and calculate $d(X', R_\sigma)$.
Step 10. Repeat Steps 4 through 9 until $\tilde{\tilde{X}}$ runs over all classes of similar equivalent standard forms. And find X^σ in all X' such that $d(X^\sigma, R_\sigma)$ is the smallest in all $d(X', R_\sigma)$. Then go to Step 3.
Step 11. Repeat Steps 3 through 10 until σ runs over all elements in S_n. And find $X^\#$ in all X^σ $(\sigma \in S_n)$ and $\sigma^\# \in S_n$ such that $d(X^\#, R_{\sigma^\#}) = \inf_{\sigma\in S_n} d(X^\sigma, R_\sigma)$.
Step 12. Calculate $R^\# = X^\#_{(\sigma^\#)^{-1}}$. Then $d(R^\#, R) = \inf_{X\in X_n} d(X, R)$.

The above FCMBP clustering method is an effective method based on the resolution structure, which contains a double loop procedure. In the external loop, σ runs over all elements in S_n. For a certain $\sigma \in S_n$, Steps 4–9 constitute the inner loop, in which $\tilde{\tilde{X}}$ runs over all classes of similar equivalent standard forms in $\tilde{X}_n/\approx$. After running over all the combination

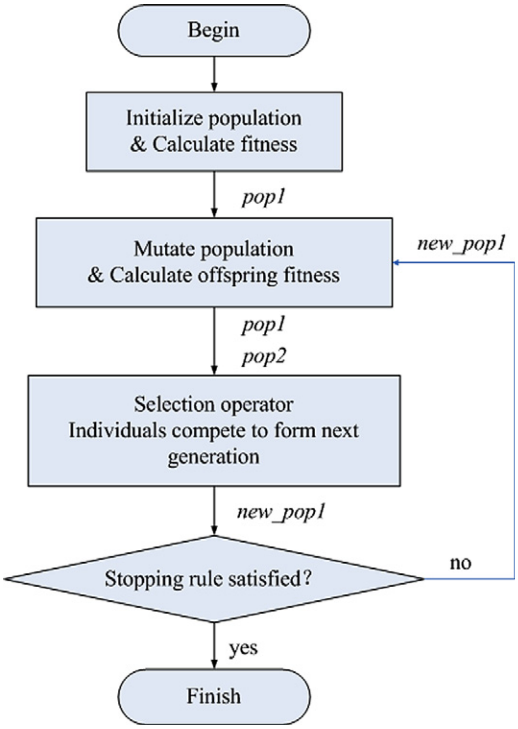


Fig. 1. Flowchart of evolutionary programming.

possibilities of all classes of similar equivalent standard forms and all elements in S_n, FCMBP fuzzy clustering method obtains the optimal fuzzy equivalent matrix from those whose parameter systems satisfy the inequality constraints.

Consider the size of S_n and $\tilde{X}_n/\approx$. For a n-order matrix, the size of S_n is $n!$.

Theorem 8 ([5]). *Let $\tilde{\tilde{k}}(n)$ be the size of $\tilde{X}_n/\approx$, then*

(1) $\tilde{\tilde{k}}(1) = 1$;
(2)

$$\tilde{\tilde{k}}(n) = \left| \sum_{m=1}^{[n/2]} \tilde{\tilde{k}}(m)\tilde{\tilde{k}}(n-m) \right|. \tag{5}$$

According to the above analysis, FCMBP method needs to run over all the $t = n!^*\tilde{\tilde{k}}(n)$ possibilities. This number grows exponentially as the order of the raw similar matrix rises, which brings difficulty in applying FCMBP method on high-order clustering problems.

3. EP-FCMBP fuzzy clustering method

Evolutionary programming is a probabilistic algorithm which maintains a population of individuals. Each individual represents a potential solution to the problem and is implemented as some data structure. A new population is formed from an existing population through the use of a mutation operator. This operator perturbs each member in the population by a random amount to produce new solutions. The degree of optimality of each individual is measured by its fitness which can be defined as a function of the cost or objective function of the problem. Through the use of a competition scheme, the individuals in each population compete with each other. The winning individuals will form a resultant population which is regarded as the next generation. Through a number of generations, the population evolves towards the global optimal point. The EP technique is iterative and the process is terminated by a stopping rule. The rule widely used includes (a) stop after a specified number of iterations, (b) stop when the fitness of the best solution reaches a given precision and (c) stop when there is no appreciable change in the best solution for a certain number of generations. The main stages of the EP technique including initialization, mutation and competitions are shown in the flowchart of Fig. 1.

The purpose of EP-FCMBP method is to find a fuzzy equivalent matrix which is as close as possible to the raw fuzzy similar matrix by a certain distance. In this paper, we adopt the distance defined in Eq. (3) to measure the difference between the obtained matrix and the raw matrix and thus design a fitness function to evaluate the fitness of the obtained equivalent

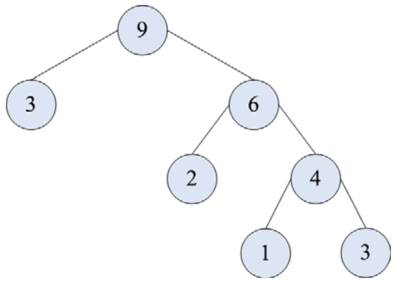


Fig. 2. Illustration of binary tree representation of an equivalent standard form.

matrix. Each feasible solution in EP-FCMBP could be directly represented by $\sigma \in S_n$ and $\tilde{\tilde{X}} \in \tilde{X}_n/\approx$. And corresponding fuzzy matrix could be obtained by Steps 5–7 in FCMBP. In order to ensure the equivalence of the obtained fuzzy matrix, EP-FCMBP resets part elements of the matrix according to some rules. Moreover, new fuzzy equivalent matrices are produced by mutating σ and $\tilde{\tilde{X}}$. This operator together with the selection mechanism in EP-FCMBP method optimizes the lack of fidelity gradually. Based on the EP methodology, an EP-FCMBP fuzzy clustering method can be established and its main components are presented as follows.

(1) Representation of solution: in FCMBP method, R_σ is obtained according to $\sigma \in S_n$ and $\tilde{\tilde{X}} \in \tilde{X}_n/\approx$. In another word, each feasible solution is determined by σ and $\tilde{\tilde{X}}$ and thus its distance with the raw fuzzy matrix could be calculated. EP-FCMBP method does not involve encoding and decoding the solutions but represent them directly. For n-order fuzzy similar matrices, permutation $\sigma \in S_n$ could be expressed by a sequence with n numbers and $\tilde{\tilde{X}} \in \tilde{X}_n/\approx$ could be represented by a corresponding binary tree. Specifically, the construction of equivalent standard forms is a recursive process. For any $X \in X_n$, which has obtained the resolution structure for X, we have $X(I_m) \in X_m$ and $X(I_m^c) \in X_{n-m}$. Then, for $X(I_m)$ and $X(I_m^c)$, we can similarly obtain the resolution structures of $X(I_m)$ and $X(I_m^c)$. This process is continued until the submatrices become one-order submatrices. The whole resolution process could be represented by a binary tree naturally. In addition, the constraint in Definition 1 that each block has more rows than columns or the rows and columns are equal makes $m \le \frac{n}{2}$ in every resolution step. Take a 9-order similar matrix for example, a permutation is expressed as $\sigma_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 4 & 5 & 3 & 8 & 1 & 2 & 6 & 7 & 9 \end{pmatrix}$, and an equivalent standard form is represented by a binary tree shown in Fig. 2. In the binary tree, the numbers in the nodes stand for the orders of the matrices to be resolved. For every non-leaf node, its left child node represents $X(I_m) \in X_m$ while its right child node corresponds to $X(I_m^c) \in X_{n-m}$. Matrices with 2 or 3 orders appear as leaf nodes in the figure because they have only one resolution structure.

(2) Initialization of population: individuals in the first generation are produced in the following way.

Step 1. Generate a permutation $\sigma \in S_n$ randomly.

Step 2. Calculate R_σ.

Step 3. Generate a random equivalent standard form by a recursion process. For every n-order matrix to be resolved in the resolution process, randomly select an integer number ranged between 1 and $\lfloor n/2 \rfloor$ as the order of its left submatrix. Take the case illustrated in Fig. 2 as an example, we explain the resolution process of building an equivalent standard form. In the first resolution step, randomly select an integer number from 1, 2, 3 and 4. This number is 3 in the case, which means the orders of the two submatrices are 3 and 6 respectively. Then, for these two submatrices, we can similarly obtain their resolution structures. A 3-order matrix has only one resolution structure, so it can be resolved into one-order matrices without generating random numbers. For the 6-order matrix, randomly select an integer number from 1, 2 and 3. The case in Fig. 2 resolves it into a 2-order submatrix and a 4-order submatrix. The whole process is continued until the submatrices become one-order submatrices. And the equivalent standard form $\tilde{\tilde{X}} \in \tilde{X}_n/\approx$ can be obtained by this process.

Step 4. Find $b_{i1}, b_{i2}, \ldots, b_{im_i}$ $(i = 1, 2, \ldots, n-1)$ in R_σ corresponding to t_i in $\tilde{\tilde{X}}$.

Step 5. Calculate $t_i' = \frac{b_{i1}+b_{i2}+\cdots+b_{im_i}}{m_i}$, $i = 1, 2, \ldots, n-1$.

The obtained t_i' does not guarantee to satisfy the inequalities given by $\tilde{\tilde{X}}$. In other words, the equivalence of the obtained matrix could not be guaranteed. At first, we treat these matrices as unfeasible solutions and tentatively set their fitness value 0. However, the primary experiments show that the above method has a very high probability of generating unfeasible solutions especially when the order of the fuzzy similar matrix is high. Too many individuals in the population have fitness 0 make the algorithm hard to converge into an optimal solution quickly. Therefore, EP-FCMBP method resets some elements in parameter system t_i' through a Step 6, which guarantees to satisfy the inequalities given by $\tilde{\tilde{X}}$ (i.e., the elements in the upper blocks are bigger than or equal to the ones in the lower blocks and the elements in the right blocks are bigger than or equal to the ones in the left blocks).

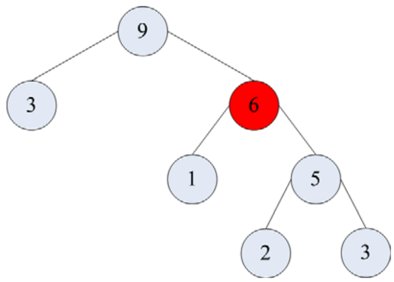


Fig. 3. Illustration of the mutation of an equivalent standard form.

Step 6. Let $t_i'' = \min(t_i', \min(t_{up}'), \min(t_{right}'))$, where t_{up}' represents all parameters in the upper blocks of t_i' and t_{right}' represents parameters in the right blocks of t_i'.

Step 7. Construct a matrix X' by using t_i'' $(i = 1, 2, \ldots, n-1)$ such that $X' \in \tilde{\tilde{X}}$.

It is to be noted that Step 6 does not change permutation σ and equivalent standard form $\tilde{\tilde{X}}$. t_i' can still be represented by the combination of σ and $\tilde{\tilde{X}}$.

(3) Fitness of candidate solutions: fitness function is used to evaluate each candidate solution's degree of satisfaction to be the best solution. The objective of EP-FCMBP method is to find a fuzzy equivalent matrix whose distance with the given fuzzy similar matrix is as small as possible. Therefore, the fitness function here should be a function concern about the distance between the two matrices. And the smaller the distance is, the higher the fitness should be. The selection operator adopted in EP is a stochastic tournament method, which does not have non-negative or any other special constraints on the fitness. It is therefore suggested here that the fitness of a solution is reciprocal of the distance between the obtained fuzzy equivalent matrix and the raw fuzzy similar matrix. For a given fuzzy similar matrix R, the fitness of a fuzzy equivalent matrix $\overline{R}$ is evaluated in Eq. (6).

$$fitness(\overline{R}) = 1/d(R, \overline{R}) = 1 \Big/ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (a_{ij} - b_{ij}) \quad R, \overline{R} \in Y_n. \tag{6}$$

(4) Producing new solutions by mutation: a new population of solutions is produced from the existing population by mutating each individual. As mentioned above, the solutions in EP-FCMBP method could be represented by the combination of permutation and equivalent standard form. Considering that the obtained fuzzy equivalent matrix will change as long as the permutation σ changes, we mutate the permutation σ first. The mutation operator is realized through randomly selecting and permuting some bits in σ. The mutation operator respectively has a probability $p_0, p_3, p_4, p_5, p_{5+}$ to choose and permute 0 bit, 3 bits, 4 bits, 5 bits and all bits in σ. The probabilities satisfy the equation $p_0 + p_3 + p_4 + p_5 + p_{5+} = 1$. Permutation $\sigma_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 4 & 5 & 3 & 8 & 1 & 2 & 6 & 7 & 9 \end{pmatrix}$ could change into $\sigma_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 4 & 1 & 3 & 8 & 7 & 2 & 5 & 6 & 9 \end{pmatrix}$ after a4-bit mutation. In this case, the 2nd, 5th, 7th, 8th bit in σ_1 are randomly selected and the corresponding number respectively changes from 5, 1, 6, 7 to 1, 7, 5, 6. Mutation operator here allows that the number of selected bit does not change after permutation. For example, σ_1 becomes $\sigma_3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 4 & 1 & 3 & 8 & 7 & 2 & 6 & 5 & 9 \end{pmatrix}$ after a4-bit mutation, in which the 2nd, 5th, 7th, 8th bit in σ_1 are randomly selected and the corresponding number respectively changes from 5, 1, 6, 7 to 1, 7, 6, 5. The number of the 7th bit remains the same after permutation although this bit is selected.

If the permutation σ changes, we randomly regenerate an equivalent standard form to form a new individual. If σ does not change, EP-FCMBP mutates the equivalent standard form only. The method randomly selects one of the non-leaf nodes in the corresponding binary tree and rebuilds the subtree rooted from that node. Fig. 3 shows a mutation result of the binary tree illustrated in Fig. 2. In this case, the red node is selected and the corresponding resolution structure of 6-order submatrix is reconstructed.

(5) The selection operator: in this competition stage, a selection mechanism is used to produce a new population from the existing parent population and the offspring population created by mutation. The selection technique used in this paper is a kind of stochastic tournament method described in the following.

For each individual in the combined population of size $2N$, a weighted value W_i of the ith individual is calculated by the following competition.

$$W_i = \sum_{t=1}^{q} W_{i,t} \tag{7}$$

where q is the competition number given by the users; $W_{i,t}$ is either 0 for loss or 1 for win as the ith individual competes with a randomly selected (rth) individual in the combined population. The value of $W_{i,t}$ is given in the following equation.

$$W_{i,t} = \begin{cases} 1 & \text{if } f_i > f_r \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where f_r is the fitness of the randomly selected rth individual and f_i is the fitness of the ith individual. When all $2N$ individuals get their competition weights, they will be ranked in a descending order according to their corresponding value W_i. The N highest scoring candidate solutions are selected as the individuals in the next generation.

4. Experimental results

In this section, a series of experiments are carried out to validate the effectiveness of the proposed EP-FCMBP fuzzy clustering method. Firstly, utilize low-order similar matrices to test the effectiveness of EP-FCMBP in finding the optimal equivalent matrix and measure its computation efficiency through comparing with FCMBP. Secondly, for high-order similar matrices whose optimal equivalent matrix could not be obtained by FCMBP in current computing environment, evaluate EP-FCMBP's performances according to the distance between the obtained equivalent matrix and the raw similar matrix. Moreover, the effects of different parameters in mutation operator are discussed in this section.

4.1. Experiments on low-order matrices

4.1.1. Results comparison with FCMBP on low-order matrices

From the previous report on FCMBP [6], the order of the fuzzy similar matrices managed by FCMBP method is no larger than 10. We first test whether EP-FCMBP method could find the same optimal fuzzy equivalent matrix which could be obtained by FCMBP. For a given fuzzy similar matrix R, the optimal fuzzy equivalent matrix obtained by FCMBP method is denoted as $R^{\#}$ and the equivalent matrix obtained by EP-FCMBP is denoted as $\overline{R}$. Take the 10-order fuzzy similar matrix R_{10} adopted in [6] for example,

$$R_{10} = \begin{bmatrix} 1 & 0.80 & 0.62 & 0.79 & 0.10 & 0.10 & 0.67 & 0.71 & 0.64 & 0.57 \\ 0.80 & 1 & 0.59 & 0.64 & 0.09 & 0.10 & 0.51 & 0.62 & 0.51 & 0.43 \\ 0.62 & 0.59 & 1 & 0.51 & 0.12 & 0.12 & 0.47 & 0.44 & 0.44 & 0.46 \\ 0.79 & 0.64 & 0.51 & 1 & 0.09 & 0.09 & 0.69 & 0.80 & 0.82 & 0.70 \\ 0.10 & 0.09 & 0.12 & 0.09 & 1 & 0 & 0.13 & 0.07 & 0.10 & 0.13 \\ 0.10 & 0.10 & 0.12 & 0.09 & 0 & 1 & 0.14 & 0.08 & 0.10 & 0.14 \\ 0.67 & 0.51 & 0.47 & 0.69 & 0.13 & 0.14 & 1 & 0.56 & 0.74 & 0.83 \\ 0.71 & 0.62 & 0.44 & 0.80 & 0.07 & 0.08 & 0.56 & 1 & 0.76 & 0.56 \\ 0.64 & 0.51 & 0.44 & 0.82 & 0.10 & 0.10 & 0.74 & 0.76 & 1 & 0.74 \\ 0.57 & 0.43 & 0.46 & 0.70 & 0.13 & 0.14 & 0.83 & 0.56 & 0.74 & 1 \end{bmatrix},$$

we carry out EP-FCMBP method on R_{10} and get $\overline{R}_{10}$ after 3000 iterations.

$$\overline{R}_{10} = \begin{bmatrix} 1 & 0.800 & 0.504 & 0.609 & 0.092 & 0.109 & 0.609 & 0.609 & 0.609 & 0.609 \\ 0.800 & 1 & 0.504 & 0.609 & 0.092 & 0.109 & 0.609 & 0.609 & 0.609 & 0.609 \\ 0.504 & 0.504 & 1 & 0.504 & 0.092 & 0.109 & 0.504 & 0.504 & 0.504 & 0.504 \\ 0.609 & 0.609 & 0.504 & 1 & 0.092 & 0.109 & 0.665 & 0.780 & 0.820 & 0.665 \\ 0.092 & 0.092 & 0.092 & 0.092 & 1 & 0.092 & 0.092 & 0.092 & 0.092 & 0.092 \\ 0.109 & 0.109 & 0.109 & 0.109 & 0.092 & 1 & 0.109 & 0.109 & 0.109 & 0.109 \\ 0.609 & 0.609 & 0.504 & 0.665 & 0.092 & 0.109 & 1 & 0.665 & 0.665 & 0.830 \\ 0.609 & 0.609 & 0.504 & 0.780 & 0.092 & 0.109 & 0.665 & 1 & 0.780 & 0.665 \\ 0.609 & 0.609 & 0.504 & 0.820 & 0.092 & 0.109 & 0.665 & 0.780 & 1 & 0.665 \\ 0.609 & 0.609 & 0.504 & 0.665 & 0.092 & 0.109 & 0.830 & 0.665 & 0.665 & 1 \end{bmatrix} = R^{\#}_{10}.$$

4.1.2. Efficiency comparison with FCMBP on finding optimal equivalent matrix

In order to systematically measure the efficiency of EP-FCMBP on finding optimal fuzzy equivalent matrix, we generate some random fuzzy similar matrices. A random n-order fuzzy similar matrix $R_{n\times n}$ satisfies: (1) diagonal entry $a_{ii} = 1,\ i = 1, 2, \ldots, n$; (2) for any element $a_{ij},\ i = 1, 2, \ldots, n-1,\ j = i+1$, it is a random number uniformly distributed in [0, 1]; (3) $a_{ji} = a_{ij}$.

For each order n ranged from 6 to 10, we respectively select 5 matrices as the test data. For each fuzzy similar matrix, we utilize FCMBP method to obtain the corresponding optimal fuzzy equivalent matrix. The EP-FCMBP method is iterated until the optimal fuzzy equivalent matrix is obtained. We record the iteration number *generation* when the method is stopped. Then the number of feasible solution to be visited when obtaining the optimal fuzzy equivalent matrix could be calculated as $t = N \times generation = 100 \times generation$. The average search cost t_2 in Table 1 is EP-FCMBP's search number averaged by the 5 matrix problems. Moreover, the number t_1 in the second column is the needed number of feasible solution to be visited for FCMBP. t_1 could be calculated according to Eq. (5) in chapter 2. Take a 10-order matrix for example, FCMBP needs

Table 1
Efficiency comparison between EP-FCMBP and FCMBP to obtain the optimal equivalent matrix (efficiency is measured according to the needed number of trial solutions).

Matrix order	Cost of FCMBP (t_1)	Search cost of EP-FCMBP						t_2/t_1
		Matrix 1	Matrix 2	Matrix 3	Matrix 4	Matrix 5	Average (t_2)	
6	**4,320**	400	1,060	820	350	850	**696**	0.161
7	**55,440**	2,670	1,090	4,530	2,820	2,810	**2,784**	0.0502
8	**967,680**	8,020	14,430	3,740	4,180	6,190	**7,312**	0.00756
9	**17,055,360**	59,140	87,850	137,470	103,060	46,760	**86,856**	0.00509
10	**373,766,400**	60,440	291,590	174,650	323,820	52,060	**180,512**	0.00048

to visit $t_1 = 10!^*\tilde{\tilde{k}}(10) = 373{,}766{,}400$ feasible solutions to obtain the optimal equivalent matrix. It is to be noted that the parameters in EP-FCMBP method are set as follows: population size $N = 100$, $q = 10$ in the selection operator, which is a typical setting combination recommended in [12], and $p_0 = 0.15,\ p_3 = 0.15,\ p_4 = 0.15,\ p_5 = 0.15,\ p_{5+} = 0.4$ in the mutation operator. Considering the randomicity of EP-FCMBP, we run the method 10 times for each similar matrix and the results in Table 1 are averaged after 10 trials.

From Table 1 we find that: (1) in order to obtain the optimal fuzzy equivalent matrix, the search cost of EP-FCMBP (t_2) is much smaller than the cost of FCMBP (t_1); (2) the superiority of EP-FCMBP becomes increasingly apparent as the order of the matrix grows. When the order $n = 10$, the search cost of EP-FCMBP is only 4.8/10,000 of FCMBP's cost. The experimental results show that EP-FCMBP method could obtain the optimal fuzzy equivalent matrix effectively and much efficiently compared with the original FCMBP method when the order of the similar matrix is no larger than 10.

4.2. Experiments on high-order matrices

4.2.1. Comparison with the transitive closure clustering method

For high-order fuzzy similar matrix, its optimal fuzzy equivalent matrix could not be obtained by FCMBP method in current PC computing environment. So we evaluate the performance of the method according to the distance between the original similar matrix and the obtained equivalent matrix. The smaller the distance is, the less the lack of fidelity is and the better clustering result will be obtained.

We first use a 12-order fuzzy similar matrix to demonstrate the effectiveness of EP-FCMBP method. The data in the matrix stems from a real-world problem in agricultural applications [13].

$$R_{12} = \begin{bmatrix} 1.00 & 0.72 & 0.54 & 0.06 & 0.51 & 0.63 & 0.37 & 0.67 & 0.48 & 0.58 & 0.16 & 0.71 \\ 0.72 & 1.00 & 0.62 & 0.05 & 0.42 & 0.72 & 0.31 & 0.57 & 0.47 & 0.47 & 0.13 & 0.65 \\ 0.54 & 0.62 & 1.00 & 0.05 & 0.39 & 0.64 & 0.25 & 0.55 & 0.39 & 0.45 & 0.07 & 0.66 \\ 0.06 & 0.05 & 0.05 & 1.00 & 0.00 & 0.05 & 0.15 & 0.05 & 0.07 & 0.01 & 0.12 & 0.04 \\ 0.51 & 0.42 & 0.39 & 0.00 & 1.00 & 0.50 & 0.29 & 0.70 & 0.65 & 0.92 & 0.13 & 0.60 \\ 0.63 & 0.72 & 0.64 & 0.05 & 0.50 & 1.00 & 0.35 & 0.63 & 0.61 & 0.47 & 0.14 & 0.65 \\ 0.37 & 0.31 & 0.25 & 0.15 & 0.29 & 0.35 & 1.00 & 0.29 & 0.41 & 0.26 & 0.42 & 0.28 \\ 0.67 & 0.57 & 0.55 & 0.05 & 0.70 & 0.63 & 0.29 & 1.00 & 0.57 & 0.67 & 0.14 & 0.84 \\ 0.48 & 0.47 & 0.39 & 0.07 & 0.65 & 0.61 & 0.41 & 0.57 & 1.00 & 0.60 & 0.20 & 0.56 \\ 0.58 & 0.47 & 0.45 & 0.01 & 0.92 & 0.47 & 0.26 & 0.67 & 0.60 & 1.00 & 0.15 & 0.53 \\ 0.16 & 0.13 & 0.07 & 0.12 & 0.13 & 0.14 & 0.42 & 0.14 & 0.20 & 0.15 & 1.00 & 0.14 \\ 0.71 & 0.65 & 0.66 & 0.04 & 0.60 & 0.65 & 0.28 & 0.84 & 0.56 & 0.53 & 0.14 & 1.00 \end{bmatrix}.$$

We can obtain a fuzzy equivalent matrix $\overline{R}_{12}$ by using the EP-FCMBP method.

$$\overline{R}_{12} = \begin{bmatrix} 1.00 & 0.64 & 0.60 & 0.06 & 0.52 & 0.64 & 0.31 & 0.69 & 0.52 & 0.52 & 0.17 & 0.69 \\ 0.64 & 1.00 & 0.60 & 0.06 & 0.52 & 0.72 & 0.31 & 0.64 & 0.52 & 0.52 & 0.17 & 0.64 \\ 0.60 & 0.60 & 1.00 & 0.06 & 0.52 & 0.60 & 0.31 & 0.60 & 0.52 & 0.52 & 0.17 & 0.60 \\ 0.06 & 0.06 & 0.06 & 1.00 & 0.06 & 0.06 & 0.06 & 0.06 & 0.06 & 0.06 & 0.06 & 0.06 \\ 0.52 & 0.52 & 0.52 & 0.06 & 1.00 & 0.52 & 0.31 & 0.52 & 0.63 & 0.92 & 0.17 & 0.52 \\ 0.64 & 0.72 & 0.60 & 0.06 & 0.52 & 1.00 & 0.31 & 0.64 & 0.52 & 0.52 & 0.17 & 0.64 \\ 0.31 & 0.31 & 0.31 & 0.06 & 0.31 & 0.31 & 1.00 & 0.31 & 0.31 & 0.31 & 0.17 & 0.31 \\ 0.69 & 0.64 & 0.60 & 0.06 & 0.52 & 0.64 & 0.31 & 1.00 & 0.52 & 0.52 & 0.17 & 0.84 \\ 0.52 & 0.52 & 0.52 & 0.06 & 0.63 & 0.52 & 0.31 & 0.52 & 1.00 & 0.63 & 0.17 & 0.52 \\ 0.52 & 0.52 & 0.52 & 0.06 & 0.92 & 0.52 & 0.31 & 0.52 & 0.63 & 1.00 & 0.17 & 0.52 \\ 0.17 & 0.17 & 0.17 & 0.06 & 0.17 & 0.17 & 0.17 & 0.17 & 0.17 & 0.17 & 1.00 & 0.17 \\ 0.69 & 0.64 & 0.60 & 0.06 & 0.52 & 0.64 & 0.31 & 0.84 & 0.52 & 0.52 & 0.17 & 1.00 \end{bmatrix}.$$

In this case, $d(R_{12}, \overline{R}_{12}) = 0.7514$. The clustering map of $\overline{R}_{12}$ is presented in Fig. 4.

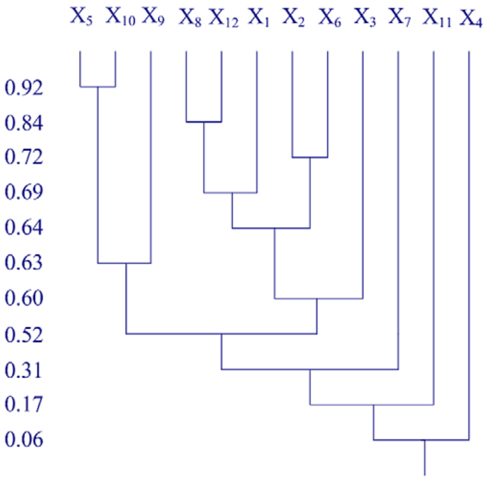


Fig. 4. EP-FCMBP clustering mapping.

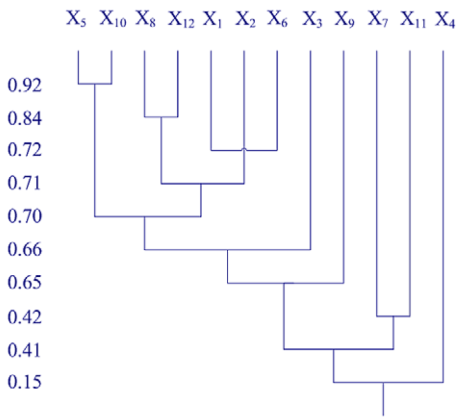


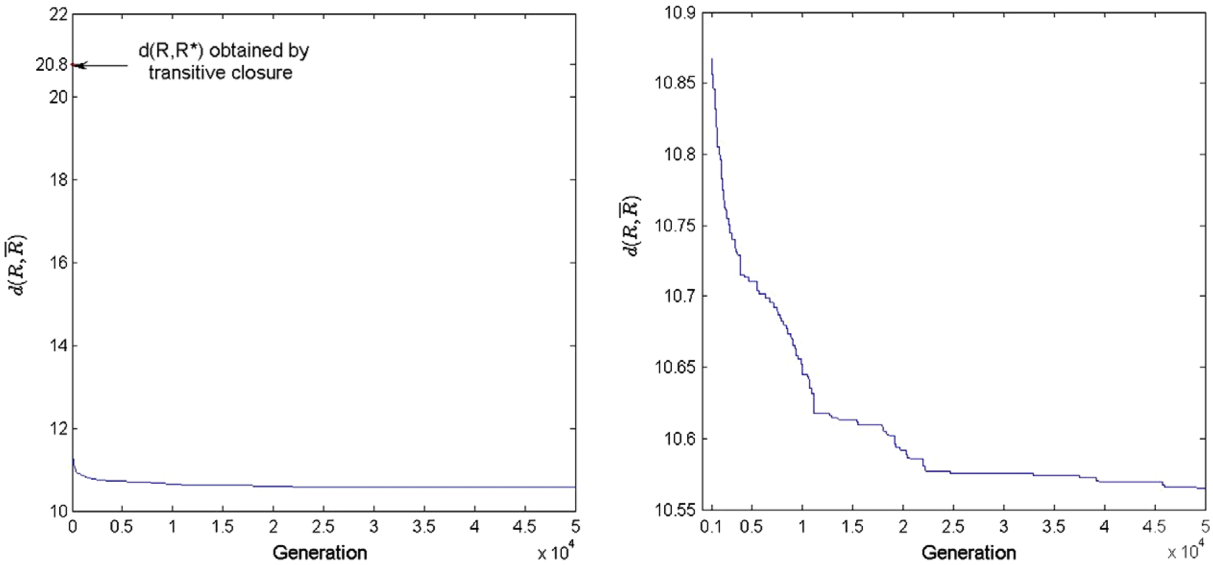
Fig. 5. Clustering map by transitive closure.

The transitive closure of R_{12} is

$$R_{12}^{*} = \begin{bmatrix} 1.00 & 0.72 & 0.66 & 0.15 & 0.70 & 0.72 & 0.41 & 0.71 & 0.65 & 0.70 & 0.41 & 0.71 \\ 0.72 & 1.00 & 0.66 & 0.15 & 0.70 & 0.72 & 0.41 & 0.71 & 0.65 & 0.70 & 0.41 & 0.71 \\ 0.66 & 0.66 & 1.00 & 0.15 & 0.66 & 0.66 & 0.41 & 0.66 & 0.65 & 0.66 & 0.41 & 0.66 \\ 0.15 & 0.15 & 0.15 & 1.00 & 0.15 & 0.15 & 0.15 & 0.15 & 0.15 & 0.15 & 0.15 & 0.15 \\ 0.70 & 0.70 & 0.66 & 0.15 & 1.00 & 0.70 & 0.41 & 0.70 & 0.65 & 0.92 & 0.41 & 0.70 \\ 0.72 & 0.72 & 0.66 & 0.15 & 0.70 & 1.00 & 0.41 & 0.71 & 0.65 & 0.70 & 0.41 & 0.71 \\ 0.41 & 0.41 & 0.41 & 0.15 & 0.41 & 0.41 & 1.00 & 0.41 & 0.41 & 0.41 & 0.42 & 0.41 \\ 0.71 & 0.71 & 0.66 & 0.15 & 0.70 & 0.71 & 0.41 & 1.00 & 0.65 & 0.70 & 0.41 & 0.84 \\ 0.65 & 0.65 & 0.65 & 0.15 & 0.65 & 0.65 & 0.41 & 0.65 & 1.00 & 0.65 & 0.41 & 0.65 \\ 0.70 & 0.70 & 0.66 & 0.15 & 0.92 & 0.70 & 0.41 & 0.70 & 0.65 & 1.00 & 0.41 & 0.70 \\ 0.41 & 0.41 & 0.41 & 0.15 & 0.41 & 0.41 & 0.42 & 0.41 & 0.41 & 0.41 & 1.00 & 0.41 \\ 0.71 & 0.71 & 0.66 & 0.15 & 0.70 & 0.71 & 0.41 & 0.84 & 0.65 & 0.70 & 0.41 & 1.00 \end{bmatrix}.$$

In this case, $d(R_{12}, R_{12}^{*}) = 1.7517$. The clustering map of R_{12}^{*} is presented in Fig. 5.

We can compare the clustering results from Figs. 4 and 5. For $\overline{R}_{12}$, and let $\lambda = 0.60$, we obtain the following fuzzy clusters from Fig. 4: $\{X_1, X_2, X_3, X_6, X_8, X_{12}\}$, $\{X_5, X_9, X_{10}\}$, $\{X_4\}$, $\{X_7\}$, $\{X_{11}\}$. On the other hand, for R_{12}^{*}, and let $\lambda = 0.66$, we obtain the following fuzzy clusters from Fig. 5: $\{X_1, X_2, X_3, X_5, X_6, X_8, X_{10}, X_{12}\}$, $\{X_4\}$, $\{X_7\}$, $\{X_9\}$, $\{X_{11}\}$. Although both the EP-FCMBP fuzzy clustering method and the transitive closure clustering method divide the twelve elements into five clusters, the clusters are quite different because $\overline{R}_{\sigma 1}$ and $R_{\sigma_2}^{*}$ are not in the same translation class. The results $d(R_{12}, \overline{R}_{12}) = 0.7514 < d(R_{12}, R_{12}^{*}) = 1.7517$ show that the EP-FCMBP method obtained a better accuracy than the transitive closure clustering method.



(a) Convergence characteristics of EP-FCMBP (all generations). (b) Convergence characteristics of EP-FCMBP (after generation 1000).

Fig. 6. Average of the minimum $d(R,\overline{R})$ at each generation over 5 trials.

Furthermore, we test the method on a 33-order fuzzy similar matrix whose data stems from another real-world application [14]. The original similar matrix R_{33} is provided in Table 2 and the equivalent matrix $\overline{R}_{33}$ obtained by EP-FCMBP method is presented in Table 3. Compared with the transitive closure R^*_{33}, $d(R_{33},\overline{R}_{33}) = 3.0828 < d(R_{33},R^*_{33}) = 3.8932$ indicates a better accuracy. And it is noted that no matrices better than $\overline{R}_{33}$ could be found so far.

In order to systematically evaluate EP-FCMBP's performance on high-order matrices, we also carry out experiments on a series of random matrices whose orders are ranged from 10 to 100. EP-FCMBP is stopped after a given iteration number and a fuzzy equivalent matrix is obtained. We measure the performance of the method according to the distance between the original similar matrix and the obtained equivalent matrix. Performance comparison between EP-FCMBP and transitive closure clustering method is presented in Table 4. $d(R,R^*)$ is the distance between the original similar matrix and the equivalent matrix obtained by the transitive closure clustering method. And $d(R,\overline{R})$ is the distance between the original similar matrix and the equivalent matrix obtained by EP-FCMBP method. The $d(R,\overline{R})$ numbers recorded in Table 4 are averaged after 5 trials. And the numbers in parentheses are standard deviations.

From the results presented in Table 4, we get the following findings: (1) high-order matrices could be managed by EP-FCMBP method, which breaks the limitation that FCMBP method could only deal with matrices whose orders are smaller than or equal to 10 in current PC computing environment. (2) In all the cases, $d(R,\overline{R}) < d(R,R^*)$ indicate that the EP-FCMBP method obtained better accuracy than the transitive method. And the t-test results with 95% confidence show that EP-FCMBP method significantly outperforms the transitive closure method. (3) The deviations are small, which indicate that EP-FCMBP is a stable method. (4) The performance of EP-FCMBP does not decline as the order of the matrices increase.

4.2.2. Characteristic of the searching process

We conduct the analysis of the evolutionary process of EP-FCMBP method. Take the process of managing a 40-order matrix for example, we averaged the minimum $d(R,\overline{R})$ value at each generation over 5 trials and plot the convergence curve in Fig. 6. Fig. 6(a) illustrates the whole process and Fig. 6(b) much clearly presents the detailed process after generation 1000. From Fig. 6(a), we can see that EP-FCMBP obtain a solution better than that obtained by transitive closure method ($d(R,R^*) = 20.776$ in this case) at the very beginning (no more than 10 generations in the worst situation over the 5 trials). Fig. 6(b) indicates that the equivalent matrix obtained by EP-FCMBP could be further optimized to gradually approach to the optimal fuzzy equivalent matrix. These features of the method endow the users with the choice to balance the running time and the precision. For some real-time demanding clustering problems, the method could obtain a solution better than that obtained by transitive closure method at a very short time. And on the other hand, for some high precision demanding clustering problems, the precision demanding could be satisfied by iterations.

4.2.3. Effect of the evolution

The original intention of this research is to break the reversal process of FCMBP by introducing EP technique. Given a fixed running time, FCMBP could stop its reversal process and output an equivalent matrix without guaranteeing the solution is the optimal equivalent matrix. Whether the introduction of evolution process has effects on finding solutions with less lack of fidelity?

Table 2
Original fuzzy similar matrix R_{33}.

1.00	0.65	0.44	0.59	0.80	0.52	0.47	0.37	0.53	0.47	0.12	0.66	0.53	0.80	0.38	0.53	0.53	0.47	0.26	0.69	0.78	0.47	0.60	0.65	0.47	0.63	0.58	0.46	0.53	0.72	0.68	0.44	0.32
0.65	1.00	0.50	0.69	0.82	0.56	0.52	0.39	0.60	0.52	0.17	0.69	0.56	0.82	0.34	0.52	0.56	0.52	0.34	0.73	0.77	0.47	0.65	0.65	0.47	0.62	0.65	0.43	0.56	0.65	0.65	0.47	0.30
0.44	0.50	1.00	0.61	0.65	0.40	0.44	0.32	0.54	0.44	0.17	0.54	0.46	0.65	0.23	0.39	0.46	0.44	0.35	0.60	0.59	0.37	0.52	0.48	0.37	0.50	0.56	0.31	0.46	0.44	0.47	0.38	0.20
0.59	0.69	0.61	1.00	0.81	0.50	0.57	0.42	0.68	0.57	0.18	0.58	0.59	0.81	0.34	0.53	0.61	0.57	0.40	0.76	0.75	0.50	0.66	0.62	0.50	0.66	0.72	0.45	0.61	0.59	0.62	0.51	0.30
0.80	0.82	0.65	0.81	1.00	0.67	0.61	0.40	0.71	0.61	0.19	0.82	0.67	0.08	0.42	0.62	0.67	0.61	0.40	0.87	0.94	0.57	0.76	0.77	0.67	0.77	0.76	0.53	0.67	0.80	0.78	0.56	0.37
0.52	0.56	0.40	0.50	0.67	1.00	0.40	0.34	0.54	0.46	0.13	0.56	0.48	0.67	0.31	0.45	0.52	0.46	0.29	0.62	0.63	0.43	0.54	0.53	0.43	0.55	0.58	0.40	0.51	0.53	0.54	0.42	0.27
0.47	0.52	0.44	0.57	0.61	0.40	1.00	0.33	0.53	0.45	0.12	0.52	0.46	0.61	0.30	0.43	0.49	0.45	0.28	0.59	0.58	0.42	0.51	0.48	0.42	0.52	0.57	0.39	0.49	0.47	0.49	0.41	0.26
0.37	0.39	0.32	0.42	0.40	0.34	0.33	1.00	0.39	0.33	0.09	0.39	0.34	0.46	0.23	0.32	0.37	0.33	0.20	0.44	0.44	0.31	0.38	0.37	0.31	0.30	0.42	0.29	0.37	0.37	0.38	0.30	0.19
0.53	0.60	0.54	0.68	0.71	0.54	0.53	0.39	1.00	0.53	0.15	0.59	0.54	0.71	0.32	0.48	0.57	0.53	0.35	0.69	0.67	0.47	0.58	0.55	0.47	0.59	0.67	0.43	0.57	0.53	0.56	0.46	0.28
0.47	0.52	0.44	0.57	0.61	0.46	0.45	0.33	0.53	1.00	0.12	0.52	0.46	0.61	0.30	0.43	0.49	0.45	0.28	0.59	0.58	0.42	0.51	0.48	0.42	0.52	0.57	0.39	0.49	0.47	0.49	0.41	0.26
0.12	0.17	0.17	0.18	0.19	0.13	0.12	0.09	0.15	0.12	1.00	0.16	0.13	0.19	0.04	0.10	0.12	0.12	0.12	0.17	0.17	0.09	0.15	0.14	0.09	0.14	0.15	0.06	0.12	0.12	0.13	0.10	0.04
0.66	0.69	0.54	0.58	0.82	0.56	0.52	0.39	0.59	0.52	0.16	1.00	0.56	0.82	0.39	0.54	0.57	0.52	0.30	0.73	0.77	0.49	0.67	0.67	0.49	0.66	0.65	0.45	0.57	0.66	0.66	0.51	0.33
0.53	0.56	0.46	0.59	0.67	0.48	0.46	0.34	0.54	0.46	0.13	0.56	1.00	0.67	0.31	0.45	0.50	0.46	0.29	0.62	0.63	0.43	0.54	0.53	0.43	0.55	0.58	0.40	0.51	0.53	0.54	0.42	0.27
0.80	0.82	0.65	0.81	0.08	0.67	0.61	0.46	0.71	0.61	0.19	0.82	0.67	1.00	0.42	0.62	0.67	0.61	0.40	0.87	0.94	0.57	0.76	0.77	0.57	0.77	0.76	0.53	0.67	0.80	0.78	0.56	0.37
0.38	0.34	0.23	0.34	0.42	0.31	0.30	0.23	0.32	0.30	0.04	0.39	0.31	0.42	1.00	0.34	0.35	0.30	0.09	0.40	0.41	0.33	0.38	0.38	0.33	0.38	0.38	0.32	0.35	0.38	0.38	0.33	0.25
0.53	0.52	0.39	0.53	0.62	0.45	0.43	0.32	0.48	0.43	0.10	0.54	0.45	0.62	0.34	1.00	0.48	0.43	0.22	0.58	0.60	0.43	0.52	0.52	0.43	0.53	0.54	0.41	0.48	0.53	0.53	0.42	0.29
0.53	0.56	0.46	0.61	0.67	0.52	0.49	0.37	0.57	0.49	0.12	0.57	0.50	0.67	0.35	0.48	1.00	0.49	0.28	0.65	0.63	0.47	0.56	0.54	0.47	0.57	0.62	0.44	0.55	0.53	0.55	0.46	0.30
0.47	0.52	0.44	0.57	0.61	0.46	0.45	0.33	0.53	0.45	0.12	0.52	0.46	0.61	0.30	0.43	0.49	1.00	0.28	0.59	0.58	0.42	0.51	0.48	0.42	0.52	0.57	0.39	0.49	0.47	0.49	0.41	0.26
0.26	0.34	0.35	0.40	0.40	0.29	0.28	0.20	0.35	0.28	0.12	0.30	0.29	0.40	0.09	0.22	0.28	0.28	1.00	0.38	0.37	0.29	0.29	0.27	0.20	0.30	0.35	0.17	0.28	0.26	0.28	0.19	0.09
0.69	0.73	0.60	0.76	0.87	0.62	0.59	0.44	0.69	0.59	0.17	0.73	0.62	0.87	0.40	0.58	0.65	0.59	0.38	1.00	0.83	0.55	0.70	0.69	0.55	0.71	0.74	0.51	0.65	0.69	0.70	0.54	0.34
0.78	0.77	0.59	0.75	0.94	0.63	0.58	0.44	0.67	0.58	0.17	0.77	0.63	0.94	0.41	0.60	0.63	0.58	0.37	0.83	1.00	0.55	0.72	0.74	0.55	0.74	0.72	0.52	0.63	0.78	0.76	0.53	0.35
0.47	0.47	0.37	0.50	0.57	0.43	0.42	0.31	0.47	0.42	0.09	0.49	0.43	0.57	0.33	0.43	0.47	0.42	0.29	0.55	0.55	1.00	0.48	0.47	0.42	0.49	0.53	0.40	0.47	0.47	0.48	0.41	0.28
0.60	0.65	0.52	0.66	0.76	0.54	0.51	0.38	0.58	0.51	0.15	0.67	0.54	0.76	0.38	0.52	0.56	0.51	0.29	0.70	0.72	0.48	1.00	0.62	0.48	0.62	0.63	0.44	0.56	0.60	0.67	0.49	0.32
0.65	0.65	0.48	0.62	0.77	0.53	0.48	0.37	0.55	0.48	0.14	0.67	0.53	0.77	0.38	0.52	0.54	0.48	0.27	0.69	0.74	0.47	0.62	1.00	0.47	0.62	0.60	0.44	0.54	0.65	0.63	0.47	0.32
0.47	0.47	0.37	0.50	0.67	0.43	0.42	0.31	0.47	0.42	0.09	0.49	0.43	0.57	0.33	0.43	0.47	0.42	0.20	0.55	0.55	0.42	0.48	0.47	1.00	0.49	0.53	0.40	0.47	0.47	0.48	0.41	0.28
0.63	0.62	0.50	0.66	0.77	0.55	0.52	0.30	0.59	0.52	0.14	0.66	0.55	0.77	0.38	0.53	0.57	0.52	0.30	0.71	0.74	0.49	0.62	0.62	0.49	1.00	0.65	0.46	0.57	0.63	0.63	0.48	0.32
0.58	0.65	0.56	0.72	0.76	0.58	0.57	0.42	0.67	0.57	0.15	0.65	0.58	0.76	0.38	0.54	0.62	0.57	0.35	0.74	0.72	0.53	0.63	0.60	0.53	0.65	1.00	0.48	0.62	0.58	0.61	0.52	0.32
0.46	0.43	0.31	0.45	0.53	0.40	0.39	0.29	0.43	0.39	0.06	0.45	0.40	0.53	0.32	0.41	0.44	0.39	0.17	0.51	0.52	0.40	0.44	0.44	0.40	0.46	0.48	1.00	0.44	0.46	0.46	0.38	0.27
0.53	0.56	0.46	0.61	0.67	0.51	0.49	0.37	0.57	0.49	0.12	0.57	0.51	0.67	0.35	0.48	0.55	0.49	0.28	0.65	0.63	0.47	0.56	0.54	0.47	0.57	0.62	0.44	1.00	0.53	0.44	0.46	0.30
0.72	0.65	0.44	0.59	0.80	0.53	0.47	0.37	0.53	0.47	0.12	0.66	0.53	0.80	0.38	0.53	0.53	0.47	0.26	0.69	0.78	0.47	0.60	0.65	0.47	0.63	0.58	0.46	0.53	1.00	0.46	0.44	0.32
0.68	0.65	0.47	0.62	0.78	0.54	0.49	0.38	0.56	0.49	0.13	0.66	0.54	0.78	0.38	0.53	0.55	0.49	0.28	0.70	0.76	0.48	0.67	0.63	0.48	0.63	0.61	0.46	0.44	0.46	1.00	0.46	0.32
0.44	0.47	0.38	0.51	0.56	0.42	0.41	0.30	0.46	0.41	0.10	0.51	0.42	0.56	0.33	0.42	0.46	0.41	0.19	0.54	0.53	0.41	0.49	0.47	0.41	0.48	0.52	0.38	0.46	0.44	0.46	1.00	0.28
0.32	0.30	0.20	0.30	0.37	0.27	0.26	0.19	0.28	0.26	0.04	0.33	0.27	0.37	0.25	0.29	0.30	0.26	0.09	0.34	0.35	0.28	0.32	0.32	0.28	0.32	0.32	0.27	0.30	0.32	0.32	0.28	1.00

Table 3
Fuzzy equivalent matrix $\overline{R}_{33}$ obtained by EP-FCMBP method.

1.00	0.63	0.63	0.59	0.63	0.57	0.45	0.36	0.57	0.45	0.12	0.63	0.59	0.63	0.25	0.45	0.57	0.45	0.12	0.63	0.63	0.40	0.63	0.63	0.40	0.63	0.57	0.40	0.57	0.63	0.63	0.45	0.25
0.63	1.00	0.63	0.59	0.63	0.57	0.45	0.36	0.57	0.45	0.12	0.63	0.59	0.63	0.25	0.45	0.57	0.45	0.12	0.63	0.63	0.40	0.63	0.65	0.40	0.63	0.57	0.40	0.57	0.63	0.63	0.45	0.25
0.63	0.63	1.00	0.59	0.65	0.57	0.45	0.36	0.57	0.45	0.12	0.65	0.59	0.63	0.25	0.45	0.57	0.45	0.12	0.63	0.63	0.40	0.63	0.63	0.40	0.64	0.57	0.40	0.57	0.63	0.63	0.45	0.25
0.59	0.59	0.59	1.00	0.59	0.57	0.45	0.36	0.57	0.45	0.12	0.59	0.59	0.59	0.25	0.45	0.57	0.45	0.12	0.59	0.59	0.40	0.59	0.59	0.40	0.59	0.57	0.40	0.57	0.59	0.59	0.45	0.25
0.63	0.63	0.65	0.59	1.00	0.57	0.45	0.36	0.57	0.45	0.12	0.65	0.59	0.63	0.25	0.45	0.57	0.45	0.12	0.63	0.63	0.40	0.63	0.63	0.40	0.64	0.57	0.40	0.57	0.63	0.63	0.45	0.25
0.57	0.57	0.57	0.57	0.57	1.00	0.45	0.36	0.57	0.45	0.12	0.57	0.57	0.57	0.25	0.45	0.57	0.45	0.12	0.57	0.57	0.40	0.57	0.57	0.40	0.57	0.58	0.40	0.57	0.57	0.57	0.45	0.25
0.45	0.45	0.45	0.45	0.45	0.45	1.00	0.36	0.45	0.45	0.12	0.45	0.45	0.45	0.25	0.45	0.45	0.45	0.12	0.45	0.45	0.40	0.45	0.45	0.40	0.45	0.45	0.40	0.45	0.45	0.45	0.45	0.25
0.36	0.36	0.36	0.36	0.36	0.36	0.36	1.00	0.36	0.36	0.12	0.36	0.36	0.36	0.25	0.36	0.36	0.36	0.12	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.25
0.57	0.57	0.57	0.57	0.57	0.57	0.45	0.36	1.00	0.45	0.12	0.57	0.57	0.57	0.25	0.45	0.57	0.45	0.12	0.57	0.57	0.40	0.57	0.57	0.40	0.57	0.57	0.40	0.57	0.57	0.57	0.45	0.25
0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.36	0.45	1.00	0.12	0.45	0.45	0.45	0.25	0.45	0.45	0.45	0.12	0.45	0.45	0.40	0.45	0.45	0.40	0.45	0.45	0.40	0.45	0.45	0.45	0.45	0.25
0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	1.00	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
0.63	0.63	0.65	0.59	0.65	0.57	0.45	0.36	0.57	0.45	0.12	1.00	0.59	0.63	0.25	0.45	0.57	0.45	0.12	0.63	0.63	0.40	0.63	0.63	0.40	0.64	0.57	0.40	0.57	0.63	0.63	0.45	0.25
0.59	0.59	0.59	0.59	0.59	0.57	0.45	0.36	0.57	0.45	0.12	0.59	1.00	0.59	0.25	0.45	0.57	0.45	0.12	0.59	0.59	0.40	0.59	0.59	0.40	0.59	0.57	0.40	0.57	0.59	0.59	0.45	0.25
0.63	0.63	0.63	0.59	0.63	0.57	0.45	0.36	0.57	0.45	0.12	0.63	0.59	1.00	0.25	0.45	0.57	0.45	0.12	0.72	0.72	0.40	0.72	0.63	0.40	0.63	0.57	0.40	0.57	0.63	0.78	0.45	0.25
0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.12	0.25	0.25	0.25	1.00	0.25	0.25	0.25	0.12	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.36	0.45	0.45	0.12	0.45	0.45	0.45	0.25	1.00	0.45	0.45	0.12	0.45	0.45	0.40	0.45	0.45	0.40	0.45	0.45	0.40	0.45	0.45	0.45	0.45	0.25
0.57	0.57	0.57	0.57	0.57	0.57	0.45	0.36	0.57	0.45	0.12	0.57	0.57	0.57	0.25	0.45	1.00	0.45	0.12	0.57	0.57	0.40	0.57	0.57	0.40	0.57	0.57	0.40	0.57	0.57	0.57	0.45	0.25
0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.36	0.45	0.45	0.12	0.45	0.45	0.45	0.25	0.45	0.45	1.00	0.12	0.45	0.45	0.40	0.45	0.45	0.40	0.45	0.45	0.40	0.45	0.45	0.45	0.45	0.25
0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	1.00	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
0.63	0.63	0.63	0.59	0.63	0.57	0.45	0.36	0.57	0.45	0.12	0.63	0.59	0.72	0.25	0.45	0.57	0.45	0.12	1.00	0.83	0.40	0.72	0.63	0.40	0.63	0.57	0.40	0.57	0.63	0.72	0.45	0.25
0.63	0.63	0.63	0.59	0.63	0.57	0.45	0.36	0.57	0.45	0.12	0.63	0.59	0.72	0.25	0.45	0.57	0.45	0.12	0.83	1.00	0.40	0.72	0.63	0.40	0.63	0.57	0.40	0.57	0.63	0.72	0.45	0.25
0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.36	0.40	0.40	0.12	0.40	0.40	0.40	0.25	0.40	0.40	0.40	0.12	0.40	0.40	1.00	0.40	0.40	0.42	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.25
0.63	0.63	0.63	0.59	0.63	0.57	0.45	0.36	0.57	0.45	0.12	0.63	0.59	0.72	0.25	0.45	0.57	0.45	0.12	0.72	0.72	0.40	1.00	0.63	0.40	0.63	0.57	0.40	0.57	0.63	0.72	0.45	0.25
0.63	0.65	0.63	0.59	0.63	0.57	0.45	0.36	0.57	0.45	0.12	0.63	0.59	0.63	0.25	0.45	0.57	0.45	0.12	0.63	0.63	0.40	0.63	1.00	0.40	0.63	0.57	0.40	0.57	0.63	0.63	0.45	0.25
0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.36	0.40	0.40	0.12	0.40	0.40	0.40	0.25	0.40	0.40	0.40	0.12	0.40	0.40	0.42	0.40	0.40	1.00	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.25
0.63	0.63	0.64	0.59	0.64	0.57	0.45	0.36	0.57	0.45	0.12	0.64	0.59	0.63	0.25	0.45	0.57	0.45	0.12	0.63	0.63	0.40	0.63	0.63	0.40	1.00	0.57	0.40	0.57	0.63	0.63	0.45	0.25
0.57	0.57	0.57	0.57	0.57	0.58	0.45	0.36	0.57	0.45	0.12	0.57	0.57	0.57	0.25	0.45	0.57	0.45	0.12	0.57	0.57	0.40	0.57	0.57	0.40	0.57	1.00	0.40	0.57	0.57	0.57	0.45	0.25
0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.36	0.40	0.40	0.12	0.40	0.40	0.40	0.25	0.40	0.40	0.40	0.12	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	1.00	0.40	0.40	0.40	0.40	0.25
0.57	0.57	0.57	0.57	0.57	0.57	0.45	0.36	0.57	0.45	0.12	0.57	0.57	0.57	0.25	0.45	0.57	0.45	0.12	0.57	0.57	0.40	0.57	0.57	0.40	0.57	0.57	0.40	1.00	0.57	0.57	0.45	0.25
0.63	0.63	0.63	0.59	0.63	0.57	0.45	0.36	0.57	0.45	0.12	0.63	0.59	0.63	0.25	0.45	0.57	0.45	0.12	0.63	0.63	0.40	0.63	0.63	0.40	0.63	0.57	0.40	0.57	1.00	0.63	0.45	0.25
0.63	0.63	0.63	0.59	0.63	0.57	0.45	0.36	0.57	0.45	0.12	0.63	0.59	0.78	0.25	0.45	0.57	0.45	0.12	0.72	0.72	0.40	0.72	0.63	0.40	0.63	0.57	0.40	0.57	0.63	1.00	0.45	0.25
0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.36	0.45	0.45	0.12	0.45	0.45	0.45	0.25	0.45	0.45	0.45	0.12	0.45	0.45	0.40	0.45	0.45	0.40	0.45	0.45	0.40	0.45	0.45	0.45	1.00	0.25
0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.12	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.12	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	1.00

Table 4
Performance comparison between EP-FCMBP and transitive closure clustering method.

Order	EP-FCMBP iteration number	Random matrix 1		Random matrix 2		Random matrix 3		Average $\frac{d(R,\overline{R})}{d(R,R^*)}$
		$d(R, R^*)$	$d(R, \overline{R})$	$d(R, R^*)$	$d(R, \overline{R})$	$d(R, R^*)$	$d(R, \overline{R})$	
10	20,000	3.5028	2.0529 (0)	4.6438	2.2040 (0)	3.9478	2.0731 (0)	0.5286
15	20,000	7.0901	3.5103 (0.0168)	6.7914	3.5924 (0.0145)	6.2624	3.1498 (0.0263)	0.5090
20	20,000	9.3008	4.9572 (0.0173)	8.9304	4.9225 (0.0186)	9.6636	5.0179 (0.0214)	0.5345
30	50,000	15.1587	7.6093 (0.0573)	15.0207	7.7014 (0.0075)	15.5173	7.7139 (0.0416)	0.5039
40	50,000	20.7760	10.5619 (0.0315)	20.9903	10.6645 (0.0464)	21.2659	10.6035 (0.0549)	0.5050
60	80,000	32.5001	16.8543 (0.0397)	32.4270	16.8123 (0.0516)	32.5499	16.4257 (0.0500)	0.5139
80	80,000	44.2357	22.5630 (0.0398)	43.9450	22.4565 (0.0472)	44.8984	22.5593 (0.0398)	0.5079
100	100,000	56.1768	28.3632 (0.0437)	55.6698	28.3449 (0.0178)	55.6246	28.1457 (0.0471)	0.5067

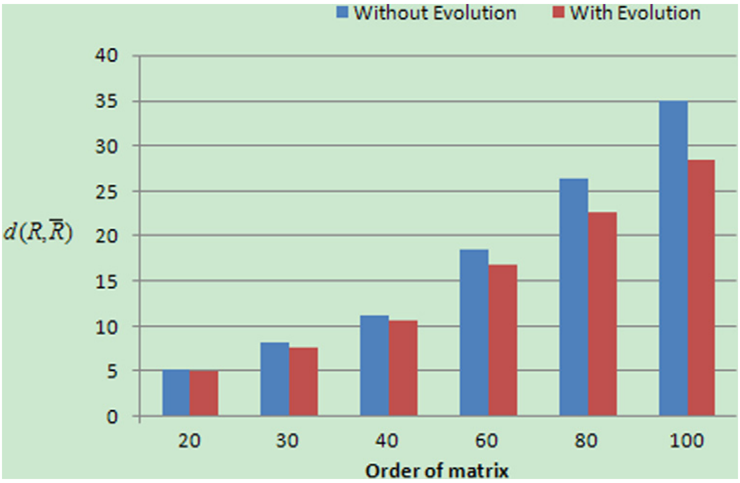


Fig. 7. Effects of the evolution mechanism in EP-FCMBP.

We carry out experiments to validate the effect of the evolution mechanism. In EP-FCMBP method, many trial solutions are generated. We endow FCMBP method with the same trial chances and record the best solution obtained by FCMBP without the selection and evolution mechanism. For each trial chance, FCMBP method randomly generates a candidate solution.

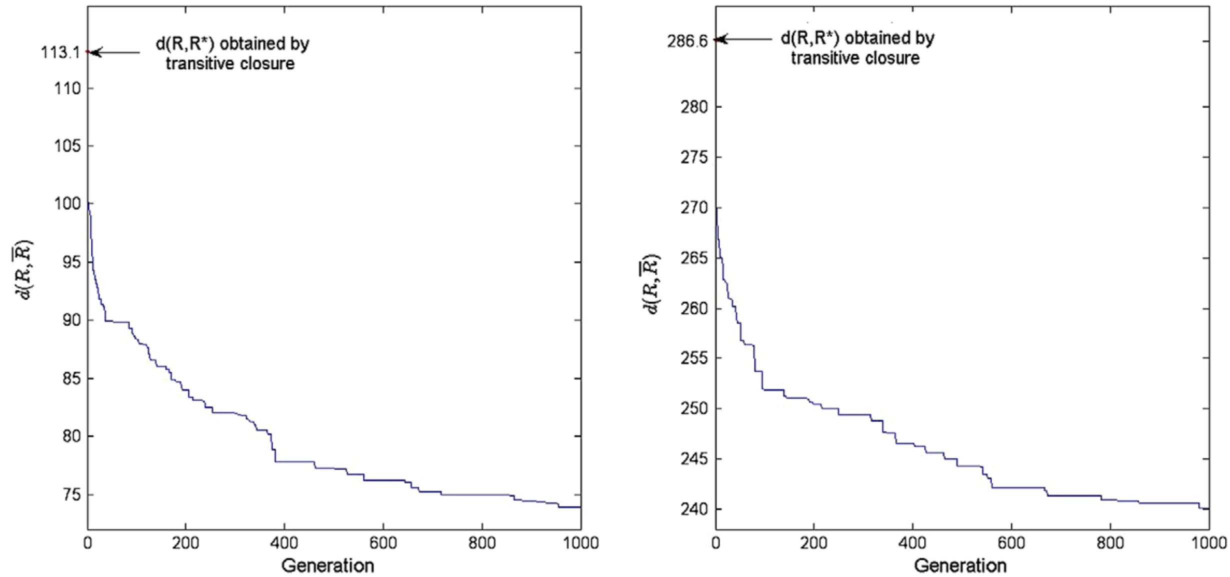
Fig. 7 compares the results of the methods with and without evolution. "With evolution" stands for the standard EP-FCMBP method and "without evolution" is the modified FCMBP method. The two methods have the same trail chances. Considering the randomicity of the method, the results illustrated in Fig. 7 are averaged after 5 trials. From Fig. 7, we can see that the results obtained by "with evolution" method have less lack of fidelity than those obtained by "without evolution" method in all 6 cases. And this superiority becomes more obvious as the order of the matrix grows. These results indicate that the evolution process in EP-FCMBP can effectively improve the quality of the solutions. And thus more accurate clustering results could be obtained.

4.2.4. Experiments on matrices with hundreds of order

Moreover, we carry out experiments on some matrices with hundreds of order. Fig. 8 gives the convergence process of EP-FCMBP operated on a 200-order matrix and a 500-order matrix respectively. From Fig. 8, we can see that EP-FCMBP obtain a solution better than that obtained by transitive closure method at the very beginning. And the solution could be further optimized through iterations to reach a higher precision demand. These results indicate that EP-FCMBP is an adoptable method for many practical problems.

4.3. Effect of different mutation parameters

In our experiments, we also analyze the performances of EP-FCMBP under different settings of the mutation parameters $p_0, p_3, p_4, p_5, p_{5+}$ on the five problems whose orders are 10, 20, 40, 60 and 100 respectively. 10 kinds of parameter settings



(a) Convergence characteristics of EP-FCMBP on a 200-order matrix. (b) Convergence characteristics of EP-FCMBP on a 500-order matrix.

Fig. 8. Convergence characteristics of EP-FCMBP on matrices with hundreds of order.

Table 5
Mutation parameter affection for performance of algorithm EP-FCMBP.

	p_0	p_3	p_4	p_5	p_{5+}	n-order problem										Average (%)
						10-order		20-order		40-order		60-order		100-order		
1	1	0	0	0	0	2.21	7.8%	5.30	6.85%	11.53	9.19%	18.76	11.3%	34.60	22%	11.4
2	0.8	0.1	0.1	0	0	2.06	0.49%	4.98	0.4%	10.65	0.85%	16.93	0.47%	28.51	0.53%	0.55
3	0.6	0.2	0.1	0.1	0	2.05	0%	4.97	0.2%	10.63	0.66%	16.99	0.83%	28.43	0.25%	0.39
4	0.5	0.2	0.1	0.1	0.1	2.05	0%	4.99	0.6%	10.55	−0.09%	16.87	0.12%	28.38	0.07%	0.14
5	0.4	0.15	0.15	0.15	0.15	2.05	0%	4.97	0.2%	10.63	0.66%	16.89	0.24%	28.37	0.04%	0.23
6	0.3	0.2	0.15	0.15	0.2	2.05	0%	4.94	−0.4%	10.63	0.66%	16.87	0.12%	28.36	0%	0.08
7	0.2	0.15	0.15	0.2	0.3	2.05	0%	4.94	−0.4%	10.55	−0.09%	16.91	0.36%	28.40	0.14%	0
8	0.2	0.1	0.15	0.15	0.4	2.05	0%	4.97	0.2%	10.55	−0.09%	16.81	−0.24%	28.35	−0.04%	−0.03
9	0.1	0.15	0.1	0.15	0.5	2.05	0%	4.97	0.2%	10.53	−0.28%	16.88	0.18%	28.40	0.14%	0.05
10	0.1	0.1	0.1	0.1	0.6	2.05	0%	4.95	−0.2%	10.60	0.38%	16.88	0.18%	28.36	0%	0.07
	0.15	0.15	0.15	0.15	0.4	2.05		4.96		10.56		16.85		28.36		0

are tested in the experiments and each result recorded in Table 5 is averaged over 5 trials. For each problem, the number in the first column is the value of $d(R, \overline{R})$, and the number in the second column is the floating ratio compared with the results obtained under the default parameter setting which is also presented in the last row of Table 5. Take the results under the 1st parameter setting on problem ID1 for example, the average $d(R, \overline{R})$ value is 2.21 over 5 trials and the floating ratio is $(2.21 - 2.05)/2.05 = 7.8\%$. And the last column in the table is the average floating ratio on the 5 problems.

From the results in Table 5, we find that EP-FCMBP's performance under the 1st parameter setting is significantly worse than those under other settings and the results obtained under the 2nd and 3rd parameter setting are little worse than that obtained under the default setting. In the 1st parameter setting, $p_0 = 1$ means that σ does not change in the mutation operator, which makes the solution searching be restricted to a small part of the solution space and most solution spaces could not be covered in this situation. In the 2nd and 3rd parameter setting, $p_{5+} = 0$ limits the bound of one mutation step which leads to the lack of fidelity little higher. Except the above three parameter settings, the average performances of EP-FCMBP method are almost the same and all present good. These results indicate that the EP-FCMBP method is insensitive to the mutation parameter setting.

5. Conclusions

This paper proposed an improved FCMBP fuzzy clustering method by introducing an evolutionary programming based optimization technique to obtain the fuzzy equivalent matrix whose distance with the raw similar matrix is as small as possible. When dealing with similar matrices whose orders are lower than or equal to ten, optimal fuzzy equivalent matrices could be obtained and the computation costs are significantly reduced compared with FCMBP. More importantly, the improved method develops FCMBP to manage similar matrices with hundreds of orders, which breaks FCMBP's computational limitation on the orders of the similar matrices in current PC computing environment. In these situations, more accurate clustering results could be obtained than that obtained by the transitive closure method and higher precision requirement could be reached by further iterations. In addition, the method has good scalability, flexibility and robustness, which is suitable in real applications.

References

[1] E.H. Ruspini, A new approach to clustering, Information and Control 15 (1969) 22–32.
[2] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenem Press, 1987.
[3] J.Y. Zhu, Some notable key problems in the applications of fuzzy clustering methods, Fuzzy Systems and Mathematics 1 (1987) 104–111.
[4] H.X. Li, Fuzzy clustering methods based on perturbation, Fuzzy Sets and Systems 33 (1989) 291–302.
[5] Q. He, H.X. Li, Fuzzy similar matrix equation $X^2 = X$ and the existence of optimal fuzzy equivalence matrix, Fuzzy Systems and Mathematics 13 (1999) 77–86.
[6] Q. He, H.X. Li, Fuzzy equivalent matrices in fuzzy clustering, Systems Engineering—Theory & Practice 19 (1999) 8–11.
[7] Q. He, S.F. Xu, J.Y. Wang, Z.Z. Shi, FCMBP clustering methods and its applications in speech recognition and fuzzy control, Journal of Systems Engineering 16 (2001) 430–437.
[8] Q. He, H.X. Li, Z.Z. Shi, E.S. Lee, Fuzzy clustering method based on perturbation, International Journal of Computers and Mathematics with Applications 46 (2003) 929–946.
[9] K.A. De Jong, Evolutionary Computation: A Unified Approach, MIT Press, MA, 2006.
[10] D.B. Fogel, Evolutionary Computation: Toward a New Philosophy in Machine Intelligence, IEEE Press, 1995.
[11] W. Gao, Comparison study of genetic algorithm and evolutionary programming, in: Proceedings of International Conference on Machine Learning and Cybernetics, 2004, pp. 204–209.
[12] T. Bäck, H.P. Schwefel, An overview of evolutionary algorithms for parameter optimization, Evolutionary Computation 1 (1993) 1–23.
[13] Y.L. Liao, Application of fuzzy clustering technique in the information analysis of sugarcane growers, Journal of Anhui Agricultural Sciences 35 (2007) 9098–9099.
[14] J.C. Han, H.K. Li, The fuzzy cluster analysis on influential factors in plant growth, Systems Engineering—Theory & Practice 8 (1999) 105–112.